UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/459,703	12/13/1999	Kiran A. Padwekar	884.027US1	1539

21186 7590 06/28/2005

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402-0938

EXAMINER	
MEONSKE, TONIA L	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 06/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/459,703	PADWEKAR, KIRAN A.	
	Examiner	Art Unit	
	Tonia L. Meonske	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 April 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2 and 4-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2 and 4-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1, 2, and 4-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Deao et al., US Patent 6,065,106.

3. Referring to claim 1, Deao et al. have taught a system comprising:

a. a storage element (Figure 45, element 4512, The storage element is where the state is saved when the processor is halted.);

b. a memory hierarchy coupled to the storage element (Figure 45, Figure 1, element 23);

c. a processor coupled to the memory hierarchy (Figure 1, element 10), wherein the processor is configured to test itself by repeatedly executing a plurality of instructions using a replay handler loaded into the memory hierarchy (Figure 45, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30).

4. Deao et al. have not specifically taught wherein the processor does not require receipt of another replay indicator to repeatedly execute the plurality of instructions. However, Kawasaki has taught a processor that does not require receipt of another replay indicator to repeatedly execute a plurality of instructions (Kawasaki, abstract, Figures 5, 6 and 8, column

6, line 61-column 7, line 20, column 7, line 49-column 9, line 24) for the desirable purpose of achieving a highly efficient programmable controller which ensures the ease of debugging. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the processor of Deao et al. not require receipt of another replay indicator to repeatedly execute the plurality of instructions, as taught by Kawasaki, for the desirable purpose of achieving a highly efficient programmable controller with optimal debugging.

5. Referring to claim 2, Deao et al. have taught the system of claim 1, as described above, and wherein the memory hierarchy is an instruction cache (Figure 1, element 23, column 8, lines 48-52).

6. Referring to claim 3, Deao et al. have taught the system of claim 1, as described above, and wherein the replay handler is loaded into the memory hierarchy in response to a signal (Figure 45, element 4512, Figure 25, Column 44, line 60 – column 45, line 43).

7. Referring to claim 4, Deao et al. have taught the system of claim 1, as described above, and wherein the replay handler includes the plurality of instructions (Figure 45, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30).

8. Referring to claim 5, Deao et al. have taught the system of claim 1, as described above, and wherein the replay handler loads the plurality of instructions into the memory hierarchy from an external cache (column 8, line 60 - column 9 line 15).

9. Referring to claim 6, Deao et al. have taught a system for replaying executions comprising:

a. a storage element (Figure 45, element 4512, The storage element is where the state is saved when the processor is halted.);

b. a memory hierarchy coupled to the storage element (Figure 45, Figure 1, element 23);

c. a system bus coupled to the memory hierarchy (Figure 1, element 41); and

d. a processor coupled to the system bus (Figure 1, element 10), the processor to execute instructions from the memory hierarchy (Figure 45, element 4510) and wherein after a replay break is received, the processor reaches to reach a steady state (Figure 45, element 4512), to transfer original code of the memory hierarchy to the storage element (Figure 45, element 4512), to load a replay handler into the memory hierarchy (Figure 45, element 4516) and to execute the replay handler to repeatedly replay at least one execution to test for proper operation of the processor, wherein at least one execution includes a plurality of instructions (Figure 45, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30).

10. Deao et al. have not specifically taught wherein the processor is capable of repeatedly replaying the at least one execution without the receipt of another replay break. However, Kawasaki has taught a processor that is capable of repeatedly replaying the at least one execution without the receipt of another replay break (Kawasaki, abstract, Figures 5, 6 and 8, column 6, line 61-column 7, line 20, column 7, line 49-column 9, line 24) for the desirable purpose of achieving a highly efficient programmable controller which ensures the ease of debugging. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the processor of Deao et al. be capable of repeatedly

replaying the at least one execution without the receipt of another replay break, as taught by Kawasaki, for the desirable purpose of achieving a highly efficient programmable controller with optimal debugging.

11. Referring to claim 7, Deao et al. have taught the system of claim 6, as described above, and wherein the original code is loaded into the memory hierarchy after the at least one execution has been repeatedly replayed (Figure 45, element 4524, column 51, line 50-column 52, line 30).

12. Referring to claim 8, Deao et al. have taught the system of claim 6, as described above, and further comprising a system memory and wherein the storage element is a location in the system memory (Figure 45, element 4512, The storage element is where the state is saved when the processor is halted. The storage element is included in the system, therefore the storage element is part of the system memory.).

13. Referring to claim 9, Deao et al. have taught the system of claim 6, as described above, but they have not specifically taught wherein the storage element in a hard drive. However, having the storage element be a hard drive allows for a greater capacity for state information to be stored and the information would be non-volatile, so the information would not be lost in the absence of power. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the system, as taught by Deao et al., have the storage element be a hard drive so that a greater amount of state information could be stored and the information would not be lost when power is absent.

14. Referring to claim 10, Deao et al. have taught a system comprising:

a. a memory hierarchy (Figure 45, Figure 1, element 23);

b. a processor coupled to the memory hierarchy wherein the processor is to execute instructions from the memory hierarchy (Figure 1, element 10);

c. a port coupled to the processor and memory hierarchy (Figure 1, The port between the test system, element 51, and the processor and the memory Heirarchy.);

d. a host system coupled to the port (Figure 1, element 51); and

e. the host system to generate a replay handler, to generate at least one execution to be repeatedly replayed by the processor when executing the handler (Figure 45, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30), and to generate a signal to the processor to cause the processor to load the replay handler into the memory hierarchy and repeatedly replay the at least one execution (Figure 45, element 4512, Figure 25, Column 44, line 60 – column 45, line 43, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30).

15. Deao et al. have not specifically taught wherein the processor is capable of repeatedly replaying the at least one execution without generation of another signal to cause the processor to load the replay handler into the memory hierarchy. However, Kawasaki has taught a processor that is capable of repeatedly replaying the at least one execution without generation of another signal to cause the processor to load the replay handler into the memory hierarchy (Kawasaki, abstract, Figures 5, 6 and 8, column 6, line 61-column 7, line 20, column 7, line 49-column 9, line 24) for the desirable purpose of achieving a highly efficient programmable controller which ensures the ease of debugging. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to

have the processor of Deao et al. be capable of repeatedly replaying the at least one execution without generation of another signal to cause the processor to load the replay handler into the memory hierarchy, as taught by Kawasaki, for the desirable purpose of achieving a highly efficient programmable controller with optimal debugging.

16. Referring to claim 11, Deao et al. have taught the system of claim 10, as described above, and wherein after the signal is generated, the processor to save original code of the memory hierarchy (Figure 45, element 4512), the processor to load the replay handler into the memory hierarchy from the host system through the port (Figure 1, Figure 45, element 4516), and the processor to execute the replay handler (Figure 45, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30).

17. Referring to claim 12, Deao et al. have taught the system of claim 11, as described above, but they have not specifically taught wherein after the replay handler is executed, the host system to modify the replay handler. However when there is a test being run, a user must be performing the test. The user would want to be able to control every aspect of the test including modifying the test so the user could extract desired information. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the system as taught by Deao et al. to have the user modify the replay handler in order to extract desired information.

18. Referring to claim 13, Deao et al. have taught the system of claim 12, as described above, but they have not taught wherein the replay handler is modified to alter starting and stopping points of one of the at least one executions. However if there is a test being run, a user must be performing the test through the host port. The user would want to be able to

control every aspect of the test including the starting and stopping points of one of the at least executions so the user could extract desired information. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the system as taught by Deao et al. to have the user modify the starting and stopping point in order to extract desired information.

19. Referring to claim 14, Deao et al. have taught the system of claim 10, as described above, and wherein a replay state is sent to the host system through the port (Figure 45, The system is being debugged, the replay states, or debug statements must be sent to the test system through the port.).

20. Referring to claim 15, Deao et al. have taught the system of claim 10, as described above, and wherein the port is a network interface (Figure 1, column 8, lines 54-56, The host debug test system (51) is connected, or networked to the processor, so the port is a network interface for debugging purposes.).

21. Referring to claim 16, Deao et al. have taught the system of claim 10, wherein the port is a serial interface (Column 55, lines 26-30).

22. Referring to claim 17, Deao et al. have taught a method for replaying executions in response to a replay signal, the replaying including:

a. interrupting normal processor execution (Figure 45, element 4512);

b. loading a replay kernel (Figure 45, element 4516);

c. repeatedly replaying at least one execution to test for proper operation of a processor, wherein the at least one execution includes a plurality of processor instructions; and

d. resuming normal executions (Figure 45, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30).

23. Deao et al. have not specifically taught wherein the repeated execution of the replay kernel does not require receipt of another replay signal. However, Kawasaki has taught wherein the repeated execution of the replay kernel does not require receipt of another replay signal (Kawasaki, abstract, Figures 5, 6 and 8, column 6, line 61-column 7, line 20, column 7, line 49-column 9, line 24) for the desirable purpose of achieving a highly efficient programmable controller which ensures the ease of debugging. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the processor of Deao et al. wherein the repeated execution of the replay kernel does not require receipt of another replay signal, as taught by Kawasaki, for the desirable purpose of achieving a highly efficient programmable controller with optimal debugging.

24. Referring to claim 18, Deao et al. have taught the method of claim 17, as described above, and further comprising generating the at least one execution (Figure 45, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30).

25. Referring to claim 19, Deao et al. have taught the method of claim 18, and further comprising accessing state information (Figure 45, elements 4512 and 4524).

26. Referring to claim 20, Deao et al. have taught a method comprising:

27. a. interrupting processes executing on a processor (Figure 45, element 4512);

b. storing minimal state information sufficient to later resume the interrupted processes (Figure 45, element 4512);

c. storing original code of an instruction cache (Figure 45, element 4512, column 51, line 50-column 52, line 30, Figure 1, element 23, column 8, lines 48-52);

d. loading a replay handler into the instruction cache(Figure 45, element 4516);

e. branching execution of the processor to the replay handler (Figure 45, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30);

f. replaying a system execution a number of times from a starting point to a stopping point while monitoring state information to test for proper operation of the processor (Figure 45, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30);

g. loading the original code into the instruction cache (Figure 45, element 4524, column 51, line 50-column 52, line 30, Figure 1, element 23, column 8, lines 48-52); and

h. resuming interrupted processes utilizing the minimal state information (Figure 45, element 4526).

28. Deao et al. have not specifically taught wherein the replaying the system execution a number of times does not require responding to another replay break. However, Kawasaki has taught wherein the replaying the system execution a number of times does not require responding to another replay break (Kawasaki, abstract, Figures 5, 6 and 8, column 6, line 61-column 7, line 20, column 7, line 49-column 9, line 24) for the desirable purpose of achieving a highly efficient programmable controller which ensures the ease of debugging. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the processor of Deao et al. wherein the replaying the system execution a number of times does not

require responding to another replay break, as taught by Kawasaki, for the desirable purpose of achieving a highly efficient programmable controller with optimal debugging.

29. Referring to claim 21, Deao et al. have taught the method of claim 20, as described above, but they have not specifically taught modifying the number of times, the starting point and the stopping point by a user. However if there is a test being run, a user must be performing the test. The user would want to be able to control every aspect of the test including the number of times and the starting and stopping points of the test so the user could extract desired information. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the system as taught by Deao et al. to have the user modify the number of times and the starting and stopping point in order to extract desired information.

30. Referring to claim 22, Deao et al. have taught the method of claim 20, as described above, and further comprising:

a. generating the system execution by tracing an execution of a program (Figure 45, Column 56, line 45 – column 45, line 10).

31. Referring to claim 23, Deao et al. have taught a computer readable medium containing computer instructions for instructing a processor to perform a method of:

a. generating at least one execution that includes a plurality of processor instructions; testing the processor in response to the processor receiving a single replay break, wherein the testing includes,

b. interrupting normal processing (Figure 45, element 4512);

c. loading a replay handler (Figure 45, element 4516);

d. repeatedly replaying the at least one execution to test for proper operation of the processor (Figure 45, elements 4516, 4518, 4520, and 4522, column 51, line 50-column 52, line 30);

e. accessing state information (Figure 45, elements 4512 and 4524);

f. storing state information (Figure 45, elements 4512 and 4524); and

g. resuming normal processing (Figure 45, element 4526).

32. Deao et al. have not specifically taught wherein the repeatedly replaying the at least one execution does not require the receipt of another replay break. However, Kawasaki has taught wherein the repeatedly replaying the at least one execution does not require the receipt of another replay break (Kawasaki, abstract, Figures 5, 6 and 8, column 6, line 61-column 7, line 20, column 7, line 49-column 9, line 24) for the desirable purpose of achieving a highly efficient programmable controller which ensures the ease of debugging. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the processor of Deao et al. wherein the repeatedly replaying the at least one execution does not require the receipt of another replay break, as taught by Kawasaki, for the desirable purpose of achieving a highly efficient programmable controller with optimal debugging.

33. Referring to claim 24, Deao et al. have taught the computer readable medium of claim 23, wherein the replay handler includes computer instructions which, when executed, cause the repeatedly replaying the at least one execution to occur (Figure 45, The system code, or computer instructions are executed. An emulation event occurs, which causes the repeatedly replaying the at least one execution to begin occurring.).

34. Referring to claim 25, Deao et al. have taught the computer readable medium of claim 23, wherein the replay handler has a predetermined number of replays for the at least one execution (Figure 45, The replay handler is predetermined to replay at least one execution.).

35. Referring to claim 26, Deao et al. have taught the computer readable medium of claim 23, wherein the replay handler dynamically determines the number of replays for the at least one execution (In Figure 45, the replay handler dynamically determines the number of replays for the at least one execution, element 4522.).

Response to Arguments

36. Applicant's arguments with respect to claims 1, 2, and 4-26 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

37. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

38. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

39. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tonia L. Meonske whose telephone number is (571) 272-4170. The examiner can normally be reached on Monday-Friday, 8-4:30.

40. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie P. Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

41. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

tlm

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100